

CLAIMS:

- Sub A<sup>1</sup> 7
- 5 steps of:
1. A method for testing a computer program comprising the  
parsing a source code of the computer program;  
creating stubs for the source code;  
instrumenting the parsed source code with the created  
stubs;  
10 compiling the instrumented code;  
testing the compiled code; and  
reporting test results in a GUI.
  2. The method of claim 1 wherein, the step of creating  
15 stubs comprises replacing the name of externally called functions  
within the source code with the name of specific functions with  
same signature as the externally called functions.
  3. The method of claim 2 wherein, the specific functions  
20 are one or more of predetermined functions and predetermined  
stubs.
  4. The method of claim 3 wherein, the predetermined  
functions and stubs are automatically generated.
  - 25 5. The method of claim 4 wherein, the step of  
automatically generating functions and stubs comprises  
automatically generating arguments to the functions and  
automatically initializing class members.
  - 30 6. The method of claim 2 wherein, the specific functions  
are user-specified functions.
  - Sub A<sup>2</sup> 7  
35 7. The method of claim 6 wherein, the user-specified  
functions are specified within the GUI.

8. The method of claim 1 further comprising the steps of breaking down the computer program into smaller components before  
5 compiling and testing the smaller components individually.

9. The method of claim 8 wherein, the smaller components are of the type of one or more of name space, class, function, and objects.

10  
Sub A<sup>3</sup> 7 10. The method of claim 1 wherein, the step of creating stubs comprises reconstructing a class by removing the source code that is not related to the class.

11. The method of claim 1 wherein, the step of creating stubs comprises reconstructing a class by ignoring the source code that is not related to the class.

12. The method of claim 2 further comprising maintaining  
20 a list of related functions to be replaced for each function under test.

13. The method of claim 1 further comprising monitoring test coverage of the computer program.

25  
Sub A<sup>4</sup> 7 14. The method of claim 13 further comprising displaying the monitored test coverage in the GUI as the test progresses.

15. The method of claim 1 further comprising the steps of  
30 defining a specific behavior when a function within the source code is called; storing the defined information; compiling the defined information as a separate object; and linking the compiled object to the code.

16. The method of claim 1 wherein, the step of testing comprises of white-box testing.

5

17. The method of claim 1 wherein, the step of testing comprises of black-box testing.

18. The method of claim 1 wherein, the step of testing  
10 comprises of regression testing.

sub A 57 19. A method for testing a computer program having a source code comprising the steps of:

15 parsing the source code of the computer program;  
breaking down the source code into a plurality of smaller components;  
testing the smaller components individually; and  
reporting test results in a GUI.

20 20. The method of claim 19 wherein, the smaller components are of the type of one or more of name space, class, function, and objects.

21. The method of claim 19 further comprising replacing the  
25 name of externally called functions within the source code with the name of specific functions with same signature as the externally called functions.

22. The method of claim 21 wherein, the specific functions  
30 are one or more of predetermined functions and predetermined stubs.

23. The method of claim 22 wherein, the predetermined functions and stubs are automatically generated.

35

24. The method of claim 23 wherein, the automatically  
generating functions and stubs comprises automatically generating  
5 arguments to the functions and automatically initializing class  
members.

25. The method of claim 21 wherein, the specific functions  
are user-specified functions.

10

26. The method of claim 19 further comprising monitoring  
test coverage of the computer program.

Sub A<sup>6</sup> 7 27. The method of claim 26 further comprising displaying  
15 the monitored test coverage in the GUI as the test progresses.

Sub B 28. The method of claim 19 further comprising the steps of  
defining a specific behavior when a function within the source  
code is called; storing the defined information; compiling the  
20 defined information as a separate object; and linking the  
compiled object to the code.

29. The method of claim 19 wherein, the step of testing  
comprises of white-box testing.

25

30. The method of claim 19 wherein, the step of testing  
comprises of black-box testing.

31. The method of claim 19 wherein, the step of testing  
30 comprises of regression testing.

Sub A<sup>7</sup> 32. A system for testing a computer program comprising:  
means for parsing a source code of the computer  
program;  
35 means for creating stubs for the source code;

Sub A77

means for instrumenting the parsed source code with the created stubs;

5 means for compiling the instrumented code;  
means for testing the compiled code; and  
means for reporting test results in a GUI.

10 33. The system of claim 32 wherein, the means for creating stubs comprises means for replacing the name of externally called functions within the source code with the name of specific functions with same signature as the externally called functions.

15 34. The system of claim 32 further comprising means for breaking down the computer program into smaller components before compiling and means for testing the smaller components individually.

20 35. The system of claim 32 further comprising means for monitoring test coverage of the computer program.

25 36. The system of claim 32 further comprising means for defining a specific behavior when a function within the source code is called; means for storing the defined information; means for compiling the defined information as a separate object; and means for linking the compiled object to the code.

30

35

